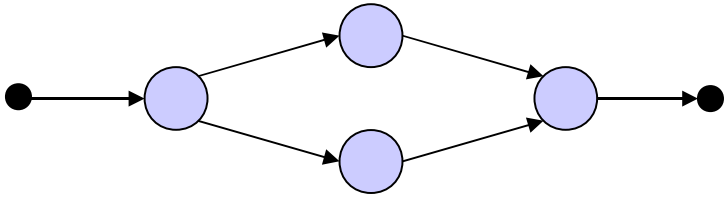
Workflow Project

Luciano Piccoli

Illinois Institute of Technology

Definitions

- Workflow
  - Description of sequence of jobs and dependencies (DAG)

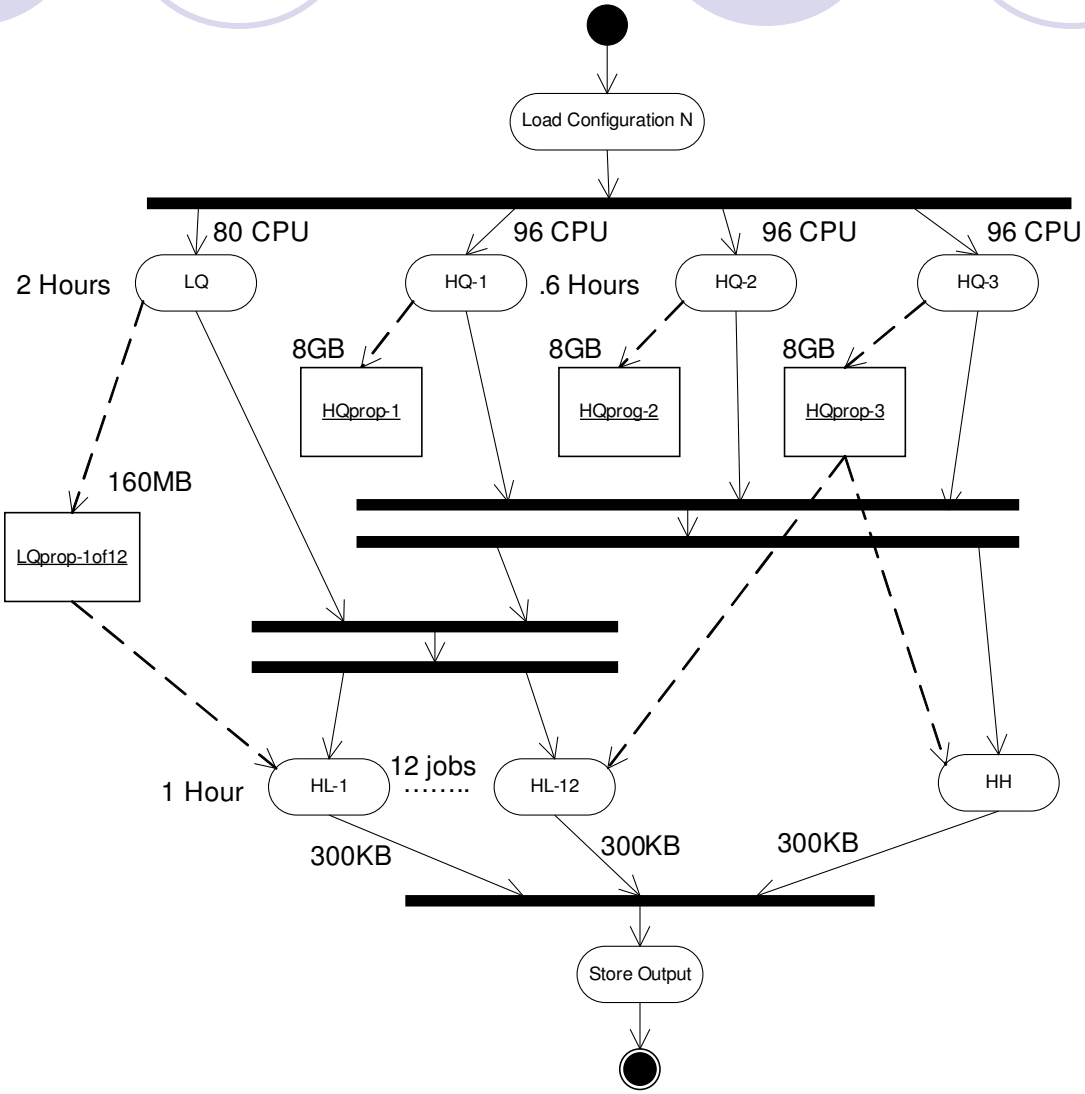


- Workflow Management
  - Coordinate a sequence of jobs based on interdependencies
  - Support parallel and distributed computation
  - Responsible for synchronizing jobs
- User focus on the work to be done rather than how jobs are mapped to machines and how to resolve dependencies

Workflow Project

- Run time environment for managing LQCD workflows
  - Integration with existing LQCD computing environments
  - Support recovery of workflows after detecting a fault
  - Keep statistics about workflow executions
  - Provide campaign monitoring
- Working with Fermilab
  - Workflow use cases from sample perl runfiles

2pt analysis campaign example



Workflow Project Milestones

- Definition of workflow system (Nov/06)
  - Under evaluation:
    - Triana (Scientific Workflow)
    - Kepler (Scientific Workflow)
    - JBoss/JBPM (Business Workflow)
- Version 1 (Jan/07)
  - Extend workflow system
    - Integration with current structure
    - Substitute for perl runfiles and bash scripts
  - Definition of interface with cluster reliability

Workflow Project Milestones

- Version 2 (May/07)
  - Ready to receive data from cluster reliability project
  - Respond to faults
    - Workflow rescheduling
    - Restart jobs (from last completed milestone)
- Version 3 (Aug/07)
  - Campaign monitoring
  - Maintain statistics on campaign execution

Workflow Project Milestones

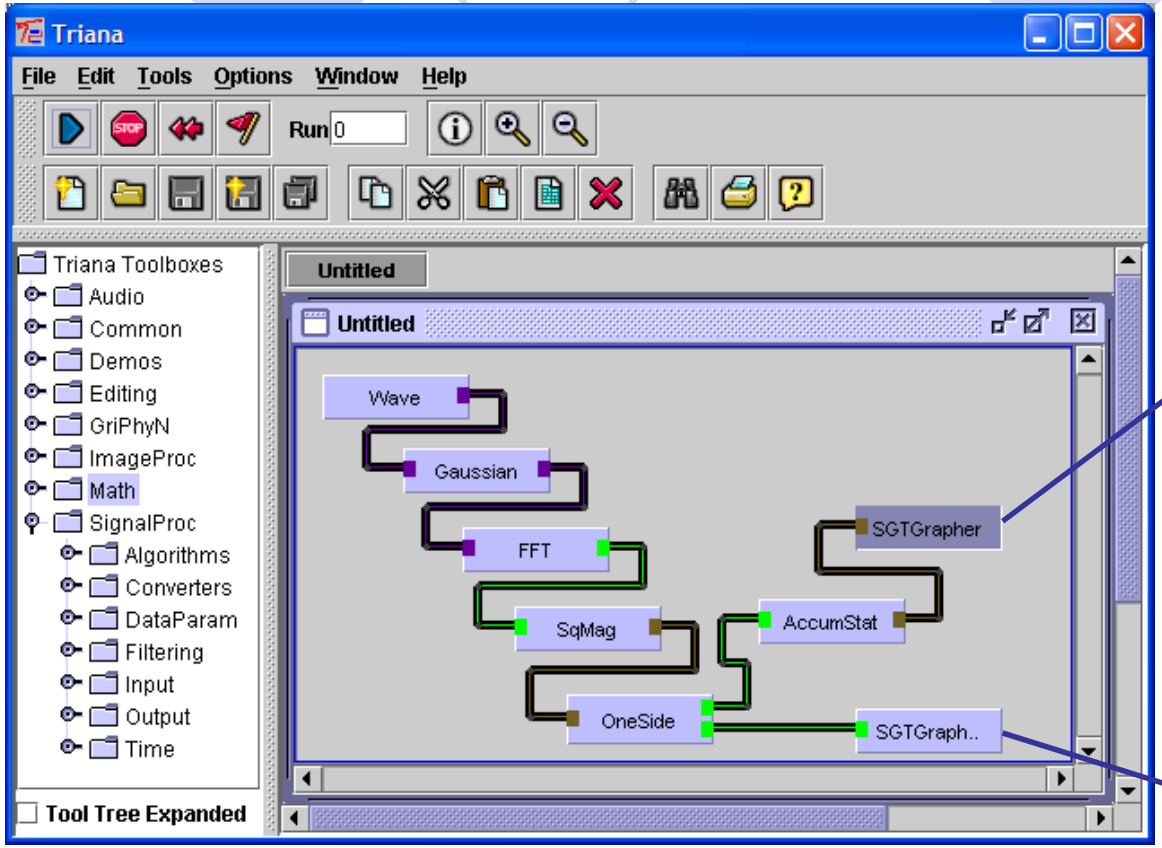
- Version 4 (Nov/07)
  - Management of intermediate files
  - Keep information on data provenance
- Version 5 (Mar/08)
  - File pre-fetching
  - Quality of service features (e.g. workflow deadline)
- Version 6 (Jun/08)
  - Multiple campaign scheduling

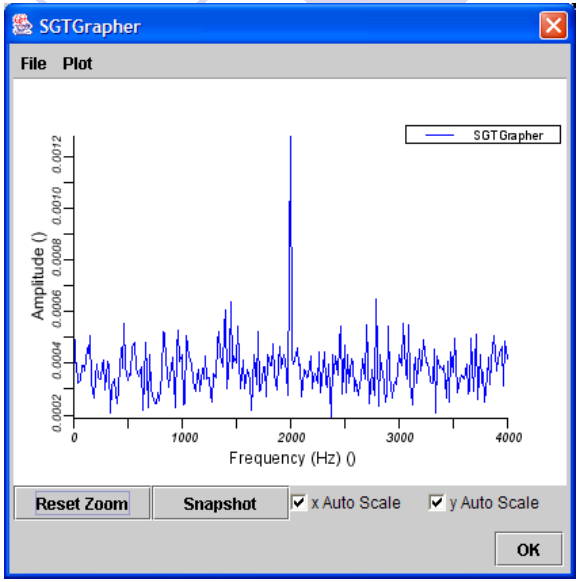
Questions?

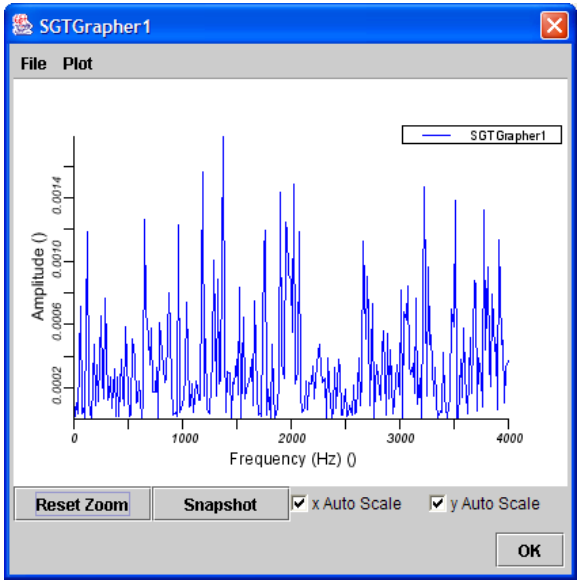
Why Workflow Management?

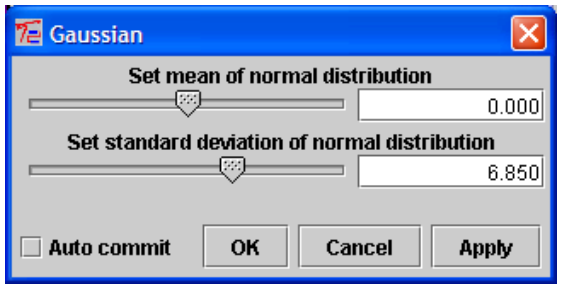
- Fault Tolerance: avoid recomputation of intermediate steps after a failure
- Portability: no changes in the workflow description when running at different sites
  - Bash scripts need to be modified
- Addresses high-level modeling of the computation rather than concentrating on low-level implementation details
- Accountability: how resources are used, who uses them

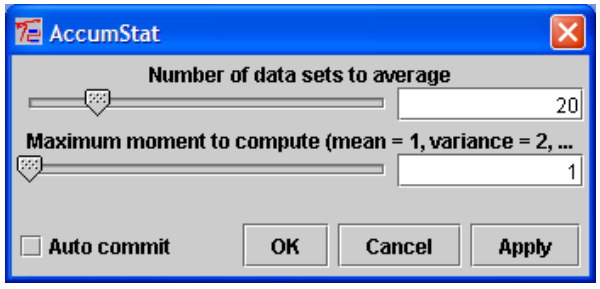
Triana


Triana
File Edit Tools Options Window Help
Run 0
Triana Toolboxes
Audio
Common
Demos
Editing
GriPhyN
ImageProc
Math
SignalProc
Algorithms
Converters
DataParam
Filtering
Input
Output
Time
Tool Tree Expanded
Untitled
Untitled
Wave
Gaussian
FFT
SqMag
OneSide
SGTGrapher
AccumStat
SGTGraph..


SGTGrapher
File Plot
SGTGrapher
Amplitude ()
Frequency (Hz) ()
Reset Zoom
Snapshot
x Auto Scale
y Auto Scale
OK


SGTGrapher1
File Plot
SGTGrapher1
Amplitude ()
Frequency (Hz) ()
Reset Zoom
Snapshot
x Auto Scale
y Auto Scale
OK


Gaussian
Set mean of normal distribution
0.000
Set standard deviation of normal distribution
6.850
Auto commit
OK
Cancel
Apply


AccumStat
Number of data sets to average
20
Maximum moment to compute (mean = 1, variance = 2, ...
1
Auto commit
OK
Cancel
Apply

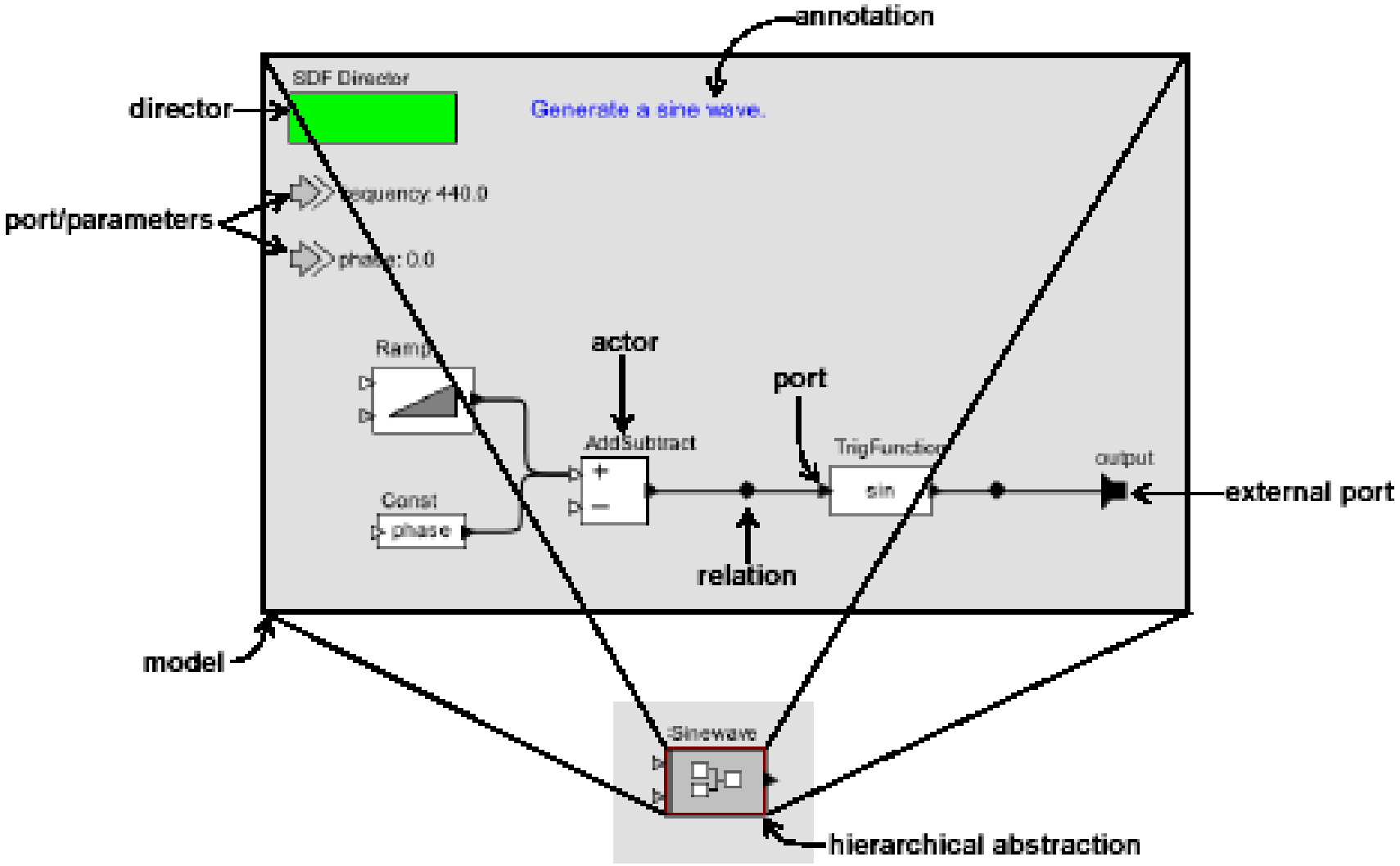
Triana Features

Language	Java
GUI	Yes
Script	Yes (Generated through the GUI)
WF Represetation	XML (Triana tags, other formats are accepted via pluggins)
Scheduling	No
External Scheduler	No
Fault Tolerance	No

Triana Features (cont)

QoS	No
Data Dependencies	Yes (component triggered by data available at input port)
Control Dependencies	Yes (by specific components, no explicit controls in the XML workflow)
Non-DAG Workflows	Yes (control loops)
Remote Execution	Yes (Triana services, through Web Services and Triana P2P to access the Grid). Tasks are executed in parallel or in pipeline.

Kepler Actor-Oriented SWF



Kepler

Language	Java
GUI	Yes (Vergil)
WF Representation	MoML (modeling markup language–an XML description of a Kepler workflow)
Concurrency control	Yes (e.g. in PN MoC)
Web services	Yes
Fault Tolerance	No
QoS constrains	No
Grid-based services	Yes
Job scheduling	Yes (using external applications, which are wrapped as commandline actors)
Data/Control dependency	Yes